COG12864TM109使用说明书

（不带字库 IC）

目　　录

1. 概述

科飞研科技专注于液晶屏及液晶模块的研发、制造。所生产COG12864TM109型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

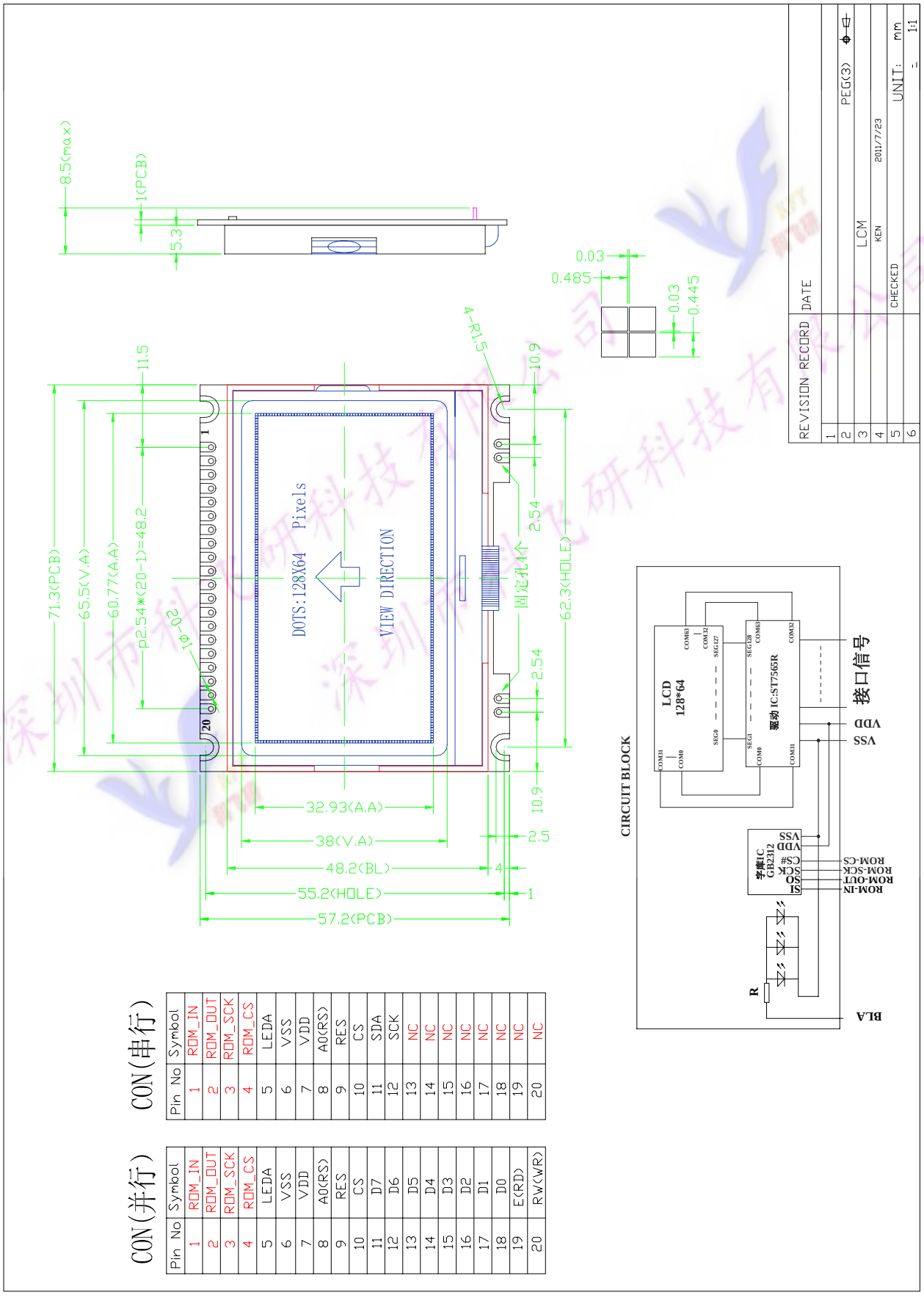
COG12864TM109可以显示128列*64行点阵单色图片，或显示8个/行*4行16*16点阵的汉字，或显示 16 个/行*8 行 8*8 点阵的英文、数字、符号。

2. COG12864TM109图像型点阵液晶模块的特性

1.1 重量轻:≤65g;
1.2 视窗大:65.5*38.0mm
1.3 结构牢：带 PCB、背光、铁框
1.4IC 采用矽创公司 ST7565R,功能强大，稳定性好
1.5 功耗低:10 - 100mW（不带背光 10mW,带背光不大于 100mW）;
1.6 显示内容:
●128*64 点阵单色图片;
●可选用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 8 字/行*4 行。按照 12*12 点阵汉字来计算可显示 10 字/行*4 行。
1.7 指令功能强:可组合成各种输入、显示、移位方式以满足不同的要求;
1.8 接口简单方便:可选串/并行接口，串行采用 4 线 SPI 串行接口，可只需 5 位 MPU 的端口(4 线 SPI 接口加上复位信号线<RESET>)。
1.9 工作温度宽:-20℃ - 70℃;
1.10 可靠性高:寿命为 50,000 小时(25℃)。

3. 外形尺寸及接口引脚功能

3.1 外形尺寸图



CON(串行)

Pin No	Symbol
1	ROM_IN
2	ROM_OUT
3	ROM_SCK
4	ROM_CS
5	LEDA
6	VSS
7	VDD
8	A0(RS)
9	RES
10	CS
11	SDA
12	SCK
13	NC
14	NC
15	NC
16	NC
17	NC
18	NC
19	NC
20	NC

CON(并行)

Pin No	Symbol
1	ROM_IN
2	ROM_OUT
3	ROM_SCK
4	ROM_CS
5	LEDA
6	VSS
7	VDD
8	A0(RS)
9	RES
10	CS
11	D7
12	D6
13	D5
14	D4
15	D3
16	D2
17	D1
18	D0
19	E(RD)
20	RW(WR)

图 1. 外形尺寸

3.2 模块的接口引脚功能

3.2.1 并行时接口引脚功能

引 线 号	符 号	名 称	功 能	
1	ROM-IN	即字库 IC 接口(SI)	串行数据输入	1.当选择带字库的产品，请参阅： （1）字库IC:GB2312说明书 （2）COG12864TM109-GB的中文字库编程说明书 2.当不用字库时为空
2	ROM-OUT	即字库 IC 接口(SO)	串行数据输出	
3	ROM-SCK	即字库 IC 接口(SCLK)	串行时钟输入	
4	ROM-CS	字库 IC 接口(CS#)	片选输入	
5	LEDA	背光电源	背光电源正极，同 VDD 电压（5V 或 3.3V）	
6	VSS	接地	0V	
7	VDD	电路电源	5V,或 3.3V 可选	
8	RS	寄存器选择信号	H:数据寄存器 0:指令寄存器（IC 资料上所写为”A0”）	
9	RES	复位	低电平复位，复位完成后，回到高电平，液晶模块开始工作	
10	CS	片选	低电平片选	
11～18	D7-D0	I/O	数据总线 DB7-DB0	
19	E（RD）	使能信号	使能信号	
20	RW(WR)	读/写	H:读数据 0:写数据	

表 1：模块并行接口引脚功能

3.2.2 串行时接口引脚功能

引 线 号	符 号	名 称	功 能	
1	ROM-IN	即字库 IC 接口(SI)	串行数据输入	1.当选择带字库的产品，请参阅： （1）字库IC:GB2312说明书 （2）COG12864TM109-GB的中文字库编程说明书 2.当不用字库时为空
2	ROM-OUT	即字库 IC 接口(SO)	串行数据输出	
3	ROM-SCK	即字库 IC 接口(SCLK)	串行时钟输入	
4	ROM-CS	字库 IC 接口(CS#)	片选输入	
5	LEDA	背光电源	背光电源正极，同 VDD 电压（5V 或 3.3V）	
6	VSS	接地	0V	
7	VDD	电路电源	5V,或 3.3V 可选	
8	RS	寄存器选择信号	H:数据寄存器 0:指令寄存器（IC 资料上所写为”A0”）	
9	RES	复位	低电平复位，复位完成后，回到高电平，液晶模块开始工作	
10	CS	片选	低电平片选	
11	SDA	I/O	串行数据	
12	SCK	I/O	串行时钟	
13～20	空	空		

表 2：模块串行接口引脚功能

4.基本原理

4.1 液晶屏（LCD）

在 LCD 上排列着 128×64 点阵,128 个列信号与驱动 IC 相连，64 个行信号也与驱动 IC 相连，IC 邦定在 LCD 玻璃上（这种加工工艺叫 COG）.

4.2 工作电图：

图1是COG12864TM109图像点阵型模块的电路框图,它由驱动IC ST7565R及几个电阻电容组成。

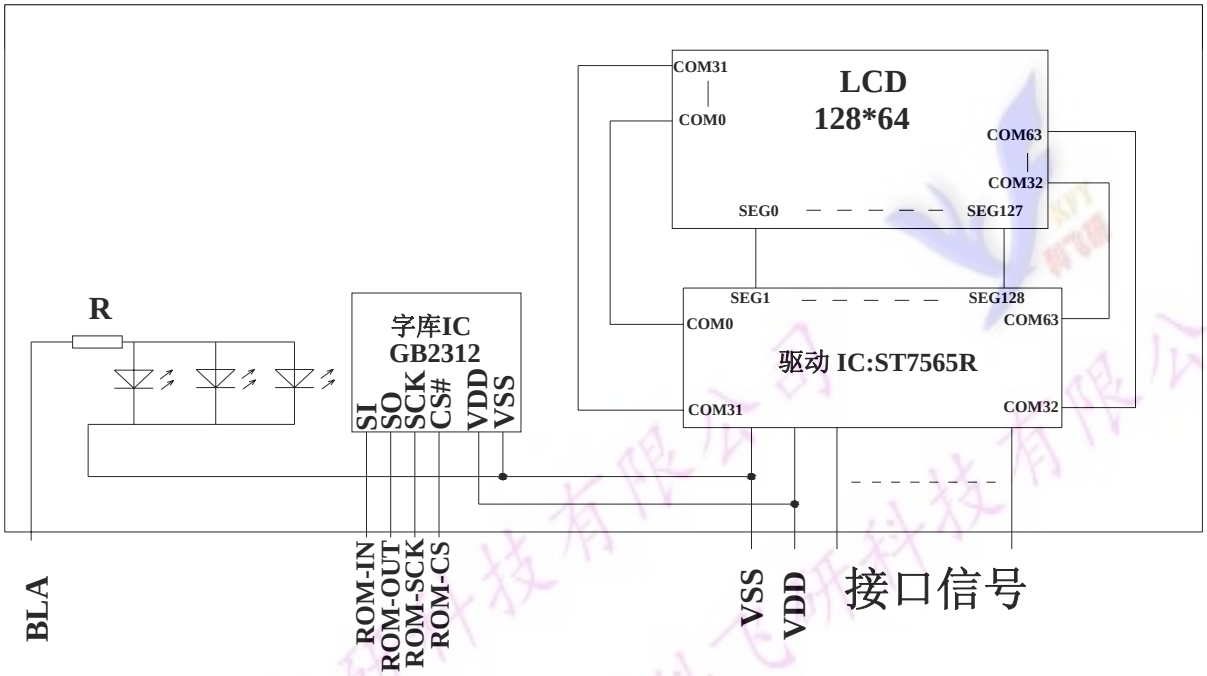


图2:COG12864TM109图像点阵型液晶模块的电路框图

4.3 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下:

工作温度:-20∽+70° C;

存储温度:-30∽+80° C;

背光板可显示绿色,黄绿色,兰色和白色。背光一般为绿色，也可为客户设计为其他颜色，但价格较绿色贵一点。

正常工作电流为：24∽60mA;

工作电压：3.0V;

正常工作条件下，LED 可连续点亮 5 万小时;

5. 技术参数

5.1 最大极限参数（超过极限参数则会损坏液晶模块）

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3		7.0	V
LCD 驱动电压	VDD - V0	VDD - 13.5		VDD + 0.3	V
静电电压		-	-	100	V
工作温度		-20		+70	℃
储存温度		-30		+80	℃

表 2：最大极限参数

5.2 直流（DC）参数

名 称	符 号	测 试 条 件	标 准 值			单位
			MIN	TYPE	MAX	

工作电压	VIN	3.3V 供电	2.4	3.3	3.6	V
		5.0V 供电	4.0	5.0	5.2	V
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIO	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOO	IOO = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.3V	-		0.3	mA
背光工作电流	ILED	VLED=3.0V	24	45	60	mA

表 3：直流（DC）参数

6. 读写时序特性

6.1 从 CPU 写到 ST7565R（Writing Data from CPU to ST7565R）

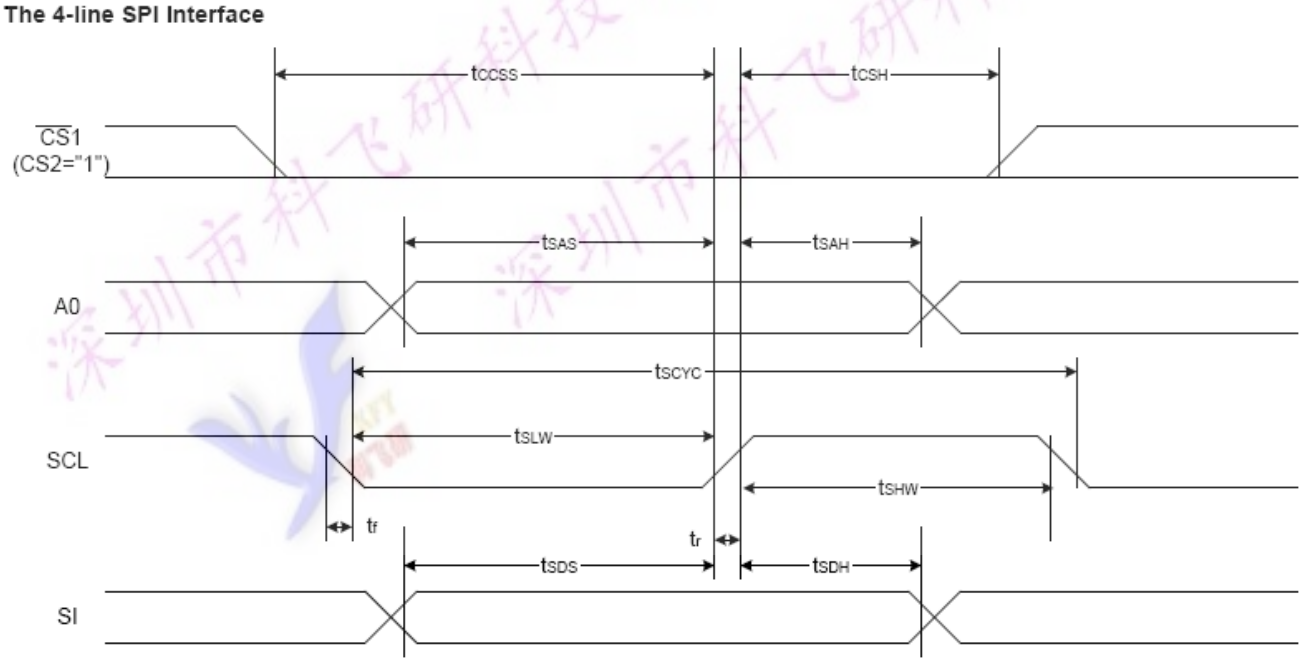


图 3. 从 CPU 写到 ST7565R（Writing Data from CPU to ST7565R

6.2 时序要求（AC 参数）：

写数据到 ST7565R 的时序要求： **表 4.**

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	

4线 SPI串口时钟周期 （4-line SPI Clock Period）	Tscyc	引脚：SCK	50	--	25	ns
保持SCK高电平脉宽 （SCK “H” pulse width）	Tshw	引脚：SCK	25			ns
保持SCK低电平脉宽 （SCK “L” pulse width）	TSLW	引脚：SCK	25			ns
地址建立时间 (Address setup time)	TSAS	引脚： RS	20	--	--	ns
地址保持时间 (Address hold time)	Tsah	引脚：RS	10	--	--	ns
数据建立时间 (Data setup time)	Tsds	引脚：SI	20	--	--	ns
数据保持时间 (Data hold time)	TSDH	引脚：SI	10	--	--	ns
片选信号建立时间 （CS-SCL time）	Tcss	引脚：CS	20			ns
片选信号保持时间 （CS-SCL time）	Tcsh	引脚：CS	40			ns

VDD =3.0V±5%, Ta = 25℃

6.3 电源启动后复位的时序要求（RESET CONDITION AFTER POWER UP）：

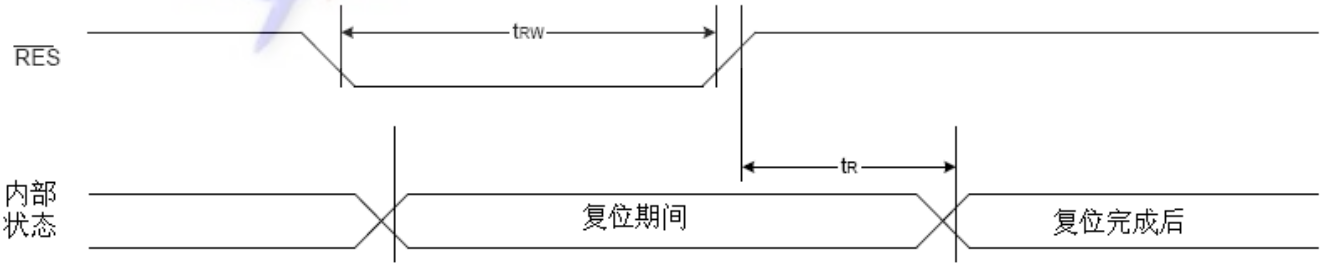


图 5：**电源启动后复位的时序**

表 6：电源启动后复位的时序要求

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	tR		--	--	1.0	us
复位保持低电平的时间	tRW	引脚：RES	1.0	--	--	us

7. 指令功能：

7.1 指令表

格式：

RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

共11种指令:1.清除,2.返回,3.输入方式设置,4.显示开关,5.控制,移位,6.功能设置,7.CGRAM地址设置,8.DDRAM地址设置,9.读忙标志,10.写数据到CG/DDRAM,11.读数据由CG/DDRAM。

指 令 表 表 8.

指令名称	指 令 码									说 明
	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(1)显示开/关（display on/off）	0	1	0	1	0	1	1	1	0 1	显示开/关: 0:关，1：开
(2)显示初始行设置（Display start line set）	0	0	1	显示初始行地址，共5位						设置显示存储器的显示初始行
(3)页地址设置 (Page address set)	0	1	0	1	1	显示页地址，共4位				设置显示页地址（注：每8行为一个页，64行分为8个页，例0000为第一页，0001为第二页
(4) 列地址高4位设置	0	0	0	0	1	列地址的高4位				高4位与低4位共同组成列地址，分别指定128列中任对应列。本液晶模块的第一列的地址为00000001，所以此指令表达为：0x10, 0x01
(4) 列地址低4位设置		0	0	0	0	列地址的低4位				
(5) 读状态（Status read）	0	状态				0	0	0	0	在本型号液晶模块不用此指令
(6)写数据(Display data write)	1	8位显示数据								从CPU写数据到液晶模块
(7)读数据(Display data read)	1	8位显示数据								在本型号液晶模块不用此指令
(8) 显示列地址增减（ADC select）		1	0	1	0	0	0	0	0 1	显示列地址增减: 0：常规：从左到右， 1：反转：从右到左
(9)显示正显/反显（Display normal/reverse）	0	1	0	1	0	0	1	1	0 1	显示正显/反显: 0:常规：正显 1:反显
(10)显示全部点阵 (Display all points)	0	1	0	1	0	0	1	0	0 1	显示全部点阵: 0:常规 1:显示全部点阵
(11)LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	0 1	设置偏压比： 0：1/9 BIAS 1：1/7BIAS
(12) Read-modify-write	0	1	1	1	0	0	0	0	0	Column address increment At write: +1 At read: 0
13) 退出上述指令(End)	0	1	1	1	0	1	1	1	0	退出上述“read/modify/write”指令

(14) 软件复位（Reset）	0	1	1	1	0	0	0	1	0	软件复位。
(15) 行扫描顺序选择(Common output mode select)		1	1	0	0	0 1	0	0	0	行扫描顺序选择： 0：普通顺序 1：反向扫描
(16) 电源控制 (Power control set)		0	0	1	0	1	电压操作模式选择，共 3 位			选择内部电压供应操作模式
(17) 选择内部电阻比例	0	0	0	1	0	0	内部电压值电阻设置			选择内部电阻比例（Rb/Ra），本液晶模块通过外置电阻设置，此指令失效
(18) 内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调，以设置液晶电压，此两个指令需紧接着使用
(18) 设置的电压值		0	0	6 位电压值数据，0～63 共 64 级						
(19)静态图标显示：开/关	0	1	0	1	0	1	1	0	0 1	0: 关, 1: 开。本液晶屏无此图标. 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)	0	1	1	1	1	1	0	0	0	选择升压倍数： 00：2 倍，3 倍，4 倍 01：5 倍 11：6 倍。本模块外部已设置升压倍数为 4 倍，不必使用此指令
		0	0	0	0	0	0	2 位数设置升压倍数		
(21) 省电模式（Power save）										省电模式，此非一条指令，是由“(10)显示全部点阵”、 (19)静态图标显示：开/关等指令合成一个“省电功能”。详细请看 IC 规格书第 47 页“POWER SAVE”
(22) 空指令（NOP）	0	1	1	1	0	0	0	1	1	空操作
(23) 测试（Test）	0	1	1	1	1	*	*	*	*	内部测试用，千万别用！

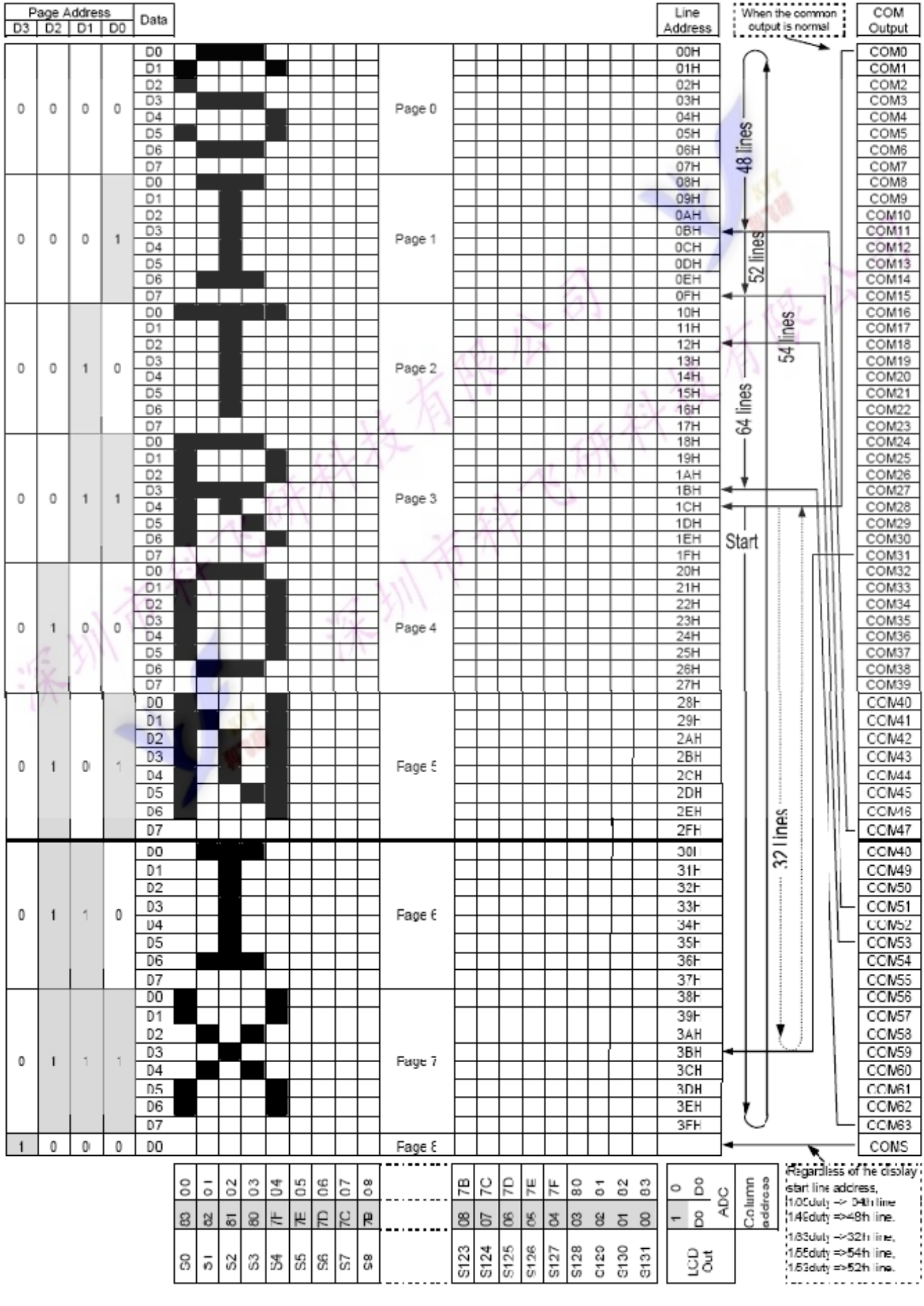
请详细参考 IC 资料”ST7564R_V15.PDF”的第 42～49 页。

7.3 点阵与 DD RAM 地址的对应关系

请留意页的定义：PAGE,与平时所讲的“页”并不是一个意思，在此表示 8 个行就是一个“页”，一个 128*64 点阵的屏分为 8 个“页”，从第 0“页”到第 7“页”。

DB7--DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。

下图摘自 ST7565R IC 资料，可通过“ST7565R_V15.PDF”之第 27 页获取最佳效果。


Page Address
D3 D2 D1 D0
Data
Line Address
When the common output is normal
COM Output
Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
48 lines
52 lines
54 lines
64 lines
32 lines
Start
COM0 COM1 COM2 COM3 COM4 COM5 COM6 COM7 COM8 COM9 COM10 COM11 COM12 COM13 COM14 COM15 COM16 COM17 COM18 COM19 COM20 COM21 COM22 COM23 COM24 COM25 COM26 COM27 COM28 COM29 COM30 COM31 COM32 COM33 COM34 COM35 COM36 COM37 COM38 COM39 COM40 COM41 COM42 COM43 COM44 COM45 COM46 COM47 COM48 COM49 COM50 COM51 COM52 COM53 COM54 COM55 COM56 COM57 COM58 COM59 COM60 COM61 COM62 COM63 COMS
S0 S1 S2 S3 S4 S5 S6 S7 S8
S123 S124 S125 S126 S127 S128 C129 S130 S131
LCD Out
ADC
Column address
Regardless of the display start line address,
1/65duty -> 64th line
1/48duty =>48th line.
1/33duty -->32th line,
1/65duty =>54th line,
1/53duty =>52th line.

7.4 初始化方法

用户所编的显示程序,开始必须进行初始化,否则模块无法正常显示,过程请参考程序

7.5 程序举例:

7.5.1 并行接口

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:

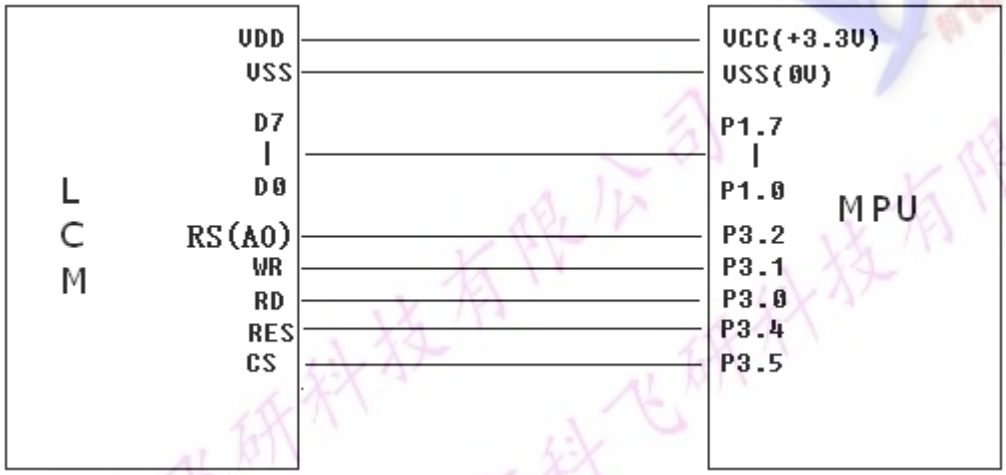


图 8.并行接口

```
/* Test program for COG12864TM109, 并行接口
   驱动 IC 是:ST7565R(or competible)
   编写,June.17th,2011

*/
#include <reg51.H>

sbit cs1=P3^4;        /*接口定义*/
sbit reset=P3^3;  /*接口定义*/
sbit rs=P3^2;     /*接口定义*/
sbit E=P3^0;      /*接口定义*/
sbit RW=P3^1;     /*接口定义。另外 P1.0~1.7 对应 DB0~DB7*/


void transfer_data(int data1);
void transfer_command(int data1);
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];
char code graphic5[];
char code graphic6[];
void delay(int i);
void Delay1(int i);
```

```
void disp_grap(char *dp);
void initial_lcd();
void clear_screen();
void waitkey();


 //===============main program==================
void main(void)
 {
   int i,j,k;
    initial_lcd();
    while(1)
    {
       clear_screen();    //clear all dots
       disp_grap(graphic1);  //display a picture of 128*64 dots
       waitkey();
       disp_grap(graphic2);  //display a picture of 128*64 dots
       waitkey();
       disp_grap(graphic4);  //display a picture of 128*64 dots
       waitkey();
       disp_grap(graphic5);  //display a picture of 128*64 dots
       waitkey();
       disp_grap(graphic6);  //display a picture of 128*64 dots
       waitkey();
    }
}

//============initial
void initial_lcd()
{
   reset=0;                    /*低电平复位*/
   delay(20);
   reset=1;                    /*复位完毕*/
   delay(20);
    transfer_command(0xe2);   /*软复位*/
    delay(5);
    transfer_command(0x2c);  /*升压步聚 1*/
    delay(5);
    transfer_command(0x2e);  /*升压步聚 2*/
    delay(5);
    transfer_command(0x2f);  /*升压步聚 3*/
    delay(5);
    transfer_command(0x23);  /*粗调对比度，可设置范围 0x20～0x27*/
    transfer_command(0x81);  /*微调对比度*/
    transfer_command(0x1f);  /*微调对比度的值，可设置范围 0x00～0x3f*/
    transfer_command(0xa2);  /*1/9 偏压比（bias）*/
```

```
    transfer_command(0xc8);  /*行扫描顺序：从上到下*/
    transfer_command(0xa0);  /*列扫描顺序：从左到右*/
    transfer_command(0x60);  /*起始行：第一行开始*/
    transfer_command(0xaf);  /*开显示*/
}



//===============clear all dot martrics=============
void clear_screen()
{
    unsigned char i,j;
    for(i=0;i<9;i++)
    {
        cs1=0;
        transfer_command(0xb0+i);
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0;j<132;j++)
        {
                transfer_data(0x00);
        }
    }
}

//==================display a piture of 128*64 dots================
void disp_grap(char *dp)
{
   int i,j;
    for(i=0;i<8;i++)
    {
        cs1=0;
        transfer_command(0xb0+i);    //set page address,
        transfer_command(0x10);
        transfer_command(0x00+1);
        for(j=0;j<128;j++)
        {
                transfer_data(*dp);
                dp++;
        }
    }
}


//=============transfer command to LCM==============
void transfer_command(int data1)
```

```
{
    cs1=0;
    rs=0;
    E=0;
    RW=0;
    P1=data1;
    E=1;
    cs1=1;
    E=0;
}

//-----------transfer data to LCM---------------
void transfer_data(int data1)
{
    cs1=0;
    rs=1;
    E=0;
    RW=0;
    P1=data1;
    E=1;
    cs1=1;
    E=0;
}

//=============delay time====================
void delay(int i)
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<990;k++);
}

//=============delay time====================
void Delay1(int i)
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<10;k++);
}

//--------------wait a switch,jump out if P2.0 get a signal"0"-----------------
void waitkey()
{
 repeat:
    if (P2&0x01) goto repeat;
    else delay(6);
```

```
    if (P2&0x01) goto repeat;
    else
    delay(40);;
}
```

7.5.2 串行接口：

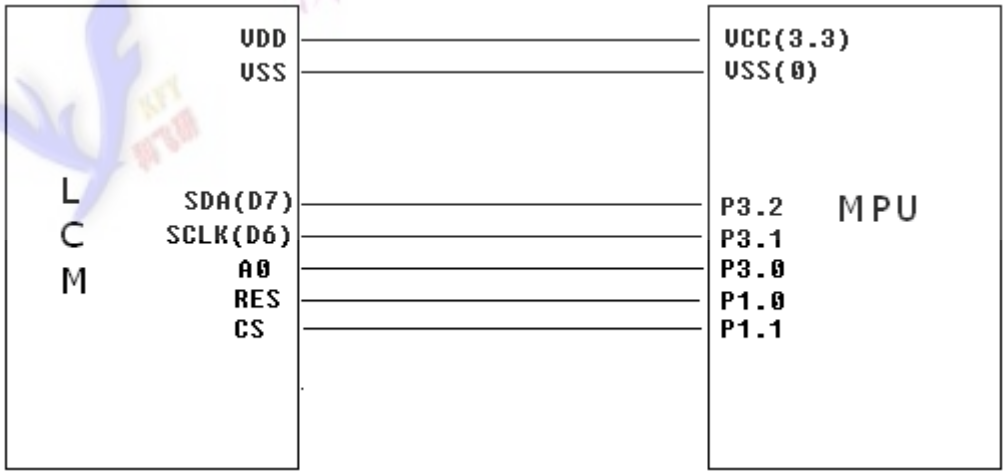


图 9.串行接口

串行程序：

```
/* Test program for COG12864TM109,串行接口
   Driver IC is:ST7565R(or competible)
   Programmed by Ken,Dec.24,2010

*/
#include <reg51.H>

sbit cs1=P1^1;
sbit reset=P1^0;
```

```
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;


void transfer_data(int data1);
void transfer_command(int data1);
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];
char code graphic5[];

void Delay(int i);
void Delay1(int i);
void disp_grap(char *dp);
void initial_lcd();
void clear_screen();
void waitkey();


 //===============main program===================
void main(void)
 {      int i,j,k;
        initial_lcd();
        while(1)
        {
              clear_screen();       //clear all dots
              disp_grap(graphic1);  //display a picture of 128*64 dots
              waitkey();
              disp_grap(graphic2);  //display a picture of 128*64 dots
              waitkey();
              disp_grap(graphic3);  //display a picture of 128*64 dots
              waitkey();
              disp_grap(graphic4);  //display a picture of 128*64 dots
              waitkey();
              disp_grap(graphic5);  //display a picture of 128*64 dots
              waitkey();
        }
}

/*LCD 初始化*/
void initial_lcd()
{
   reset=0;                    /*低电平复位*/
   delay(20);
```

```
    reset=1;                    /*复位完毕*/
    delay(20);
     transfer_command(0xe2);   /*软复位*/
     delay(5);
     transfer_command(0x2c);  /*升压步聚 1*/
     delay(5);
     transfer_command(0x2e);  /*升压步聚 2*/
     delay(5);
     transfer_command(0x2f);  /*升压步聚 3*/
     delay(5);
     transfer_command(0x23);  /*粗调对比度，可设置范围 0x20～0x27*/
     transfer_command(0x81);  /*微调对比度*/
     transfer_command(0x1f);  /*微调对比度的值，可设置范围 0x00～0x3f*/
     transfer_command(0xa2);  /*1/9 偏压比（bias）*/
     transfer_command(0xc8);  /*行扫描顺序：从上到下*/
     transfer_command(0xa0);  /*列扫描顺序：从左到右*/
     transfer_command(0x60);  /*起始行：第一行开始*/
     transfer_command(0xaf);  /*开显示*/
}
//===============clear all dot martrics============
void clear_screen()
{
     unsigned char  i,j;
    for(i=0;i<9;i++)
     {
            cs1=0;
            transfer_command(0xb0+i);
            transfer_command(0x10);
            transfer_command(0x00);
            for(j=0;j<132;j++)
            {
                transfer_data(0x00);
            }
     }
}

//=================display a piture of 128*64 dots===============
void disp_grap(char *dp)
{
      int i,j;
      for(i=0;i<8;i++)
      {
            cs1=0;
            transfer_command(0xb0+i);    //set page address,
            transfer_command(0x10);
            transfer_command(0x00);
```

```
            for(j=0;j<128;j++)
            {
                transfer_data(*dp);
                dp++;
            }
        }
}


//=============transfer command to LCM===============
void transfer_command(int data1)
{
    char i;
    cs1=0;
    rs=0;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        Delay1(5);
        sclk=1;
        Delay1(5);
        data1=data1<<=1;

    }
}

//-----------transfer data to LCM---------------
void transfer_data(int data1)
{
    char i;
    cs1=0;
    rs=1;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
     }
}

//=============delay time====================
void Delay(int i)
```

```
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<990;k++);
}

//=============delay time====================
void Delay1(int i)
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<10;k++);
}

//--------------wait a switch,jump out if P2.0 get a signal"0"-----------------
void waitkey()
{
 repeat:
    if (P2&0x01) goto repeat;
    else Delay(1);
    if (P2&0x01) goto repeat;
    else;
}
```